

Order barrier for low-storage DIRK methods with positive weights*

I. Higuera, T. Roldán

Departamento de Estadística, Informática y Matemáticas,
Universidad Pública de Navarra,
Campus de Arrosadia, 31006 Pamplona, Spain.

Abstract

In this paper we study an order barrier for low-storage diagonally implicit Runge-Kutta (DIRK) methods with positive weights. The Butcher matrix for these schemes, that can be implemented with only two memory registers in the van der Houwen implementation, has a special structure that restricts the number of free parameters of the method. We prove that third order low-storage DIRK methods must contain negative weights, obtaining the order barrier $p \leq 2$ for these schemes. This result extends the well known one for symplectic DIRK methods, which are a particular case of low-storage DIRK methods. Some other properties of second order low-storage DIRK methods are given.

Keywords Diagonally Implicit, DIRK, Runge-Kutta, low-storage, symplectic, stiff problems, time discretization, composition

Mathematics Subject Classification 65L06, 65L05, 65L04

1 Introduction and main results

Space discretization of some time-dependent partial differential equations (PDEs) gives rise to systems of ordinary differential equations (ODEs)

$$(1) \quad y' = \tilde{f}(y), \quad y(t_0) = y_0,$$

whose numerical solution is obtained with a time-stepping method. For systems (1) with a large number of equations, the high dimension of the problem compromises the computer memory capacity and thus it is important to incorporate low memory usage to some other properties of the scheme. These ideas have been developed, e.g., in [1, 7, 10, 11, 12, 16, 17], where different low-storage Runge-Kutta methods have been constructed. The most commonly used low-storage implementation techniques are the ones by van der Houwen [16] and Williamson [17].

However, when the differential system (1) contains stiff terms, severe step size restrictions arise for explicit schemes. If the differential system involves terms with different stiffness properties, an efficient

*This is a pre-print of an article published in Journal of Scientific Computing. The final authenticated version is available online at: <https://doi.org/10.1007/s10915-017-0540-6>

approach to obtain numerical approximations is the use of implicit-explicit (IMEX) time-discretizations. In this procedure, if the differential system is of the form

$$(2) \quad y' = f(y) + g(y), \quad y(t_0) = y_0,$$

an explicit method is used for the non-stiff part f and an implicit one is used for the stiff one g .

A class of IMEX methods that have been used in the literature are Additive Semi-Implicit Runge-Kutta (ASIRK) schemes [18], whose implicit part is a diagonally implicit Runge-Kutta (DIRK) method. The numerical solution of (2) with an s -stage ASIRK- sA method is given by

$$(3) \quad y_{n+1} = y_n + \sum_{i=1}^s b_i K_{i,n+1},$$

where the internal derivatives $K_{i,n+1}$ are given by

$$(4) \quad K_{i,n+1} = h \left(f(y_n + \sum_{j=1}^{i-1} \tilde{a}_{ij} K_{j,n+1}) + g(y_n + \sum_{j=1}^{i-1} a_{ij} K_{j,n+1} + a_{ii} K_{i,n+1}) \right), \quad i = 1, \dots, s,$$

and $\tilde{\mathcal{A}} = (\tilde{a}_{ij})$, $\mathcal{A} = (a_{ij})$, $b^t = (b_j)$ are the coefficients of the method. Notice that for $g = 0$, (3)-(4) are the equations of an explicit Runge-Kutta (RK) method for problem $y' = f(y)$; in a similar way, for $f = 0$, (3)-(4) are the equations of a DIRK method for problem $y' = g(y)$.

A naive implementation of a standard RK, DIRK or ASIRK- sA method requires $s+1$ memory registers of length N , where N is the dimension of the differential problem (2). Depending on the method used to solve the nonlinear systems associated to the DIRK scheme, additional registers may be required (see [8, Remark 2]).

In [8], we used the van der Houwen implementation technique to construct low-storage ASIRK- sA schemes. More precisely, we focused our attention on explicit RK and DIRK methods such that only two memory registers are needed for each scheme, and whose combination as ASIRK- sA methods requires just three memory registers. According to this, the number of registers is independent of the number of stages of the scheme and thus, by increasing the number of stages, high order methods can be constructed without increasing the number of memory registers required for their implementation.

The price to be paid for such reduction on the number of registers is a decrease on the number of free parameters in the scheme, together with a special disposal of them in the Butcher matrices. The structure of these matrices must be of the form

$$\tilde{\mathcal{A}} = \begin{pmatrix} 0 & & & & \\ b_1 + \gamma_1 & 0 & & & \\ b_1 & b_2 + \gamma_2 & 0 & & \\ \vdots & \vdots & \ddots & \ddots & \\ b_1 & b_2 & \dots & b_{s-1} + \gamma_{s-1} & 0 \end{pmatrix}, \quad \mathcal{A} = \begin{pmatrix} b_1/\alpha_1 & & & & \\ b_1 & b_2/\alpha_2 & & & \\ b_1 & b_2 & b_3/\alpha_3 & & \\ \vdots & \vdots & \ddots & \ddots & \\ b_1 & b_2 & \dots & b_{s-1} & b_s/\alpha_s \end{pmatrix},$$

where $b^t = (b_1, b_2, \dots, b_s)$ and $\alpha_i \neq 0$ for all i .

In this paper, we focus on the implicit part of the low-storage ASIRK- sA scheme. Thus, we are inter-

ested on low-storage DIRK methods (\mathcal{A}, b) that can be implemented with two memory registers,

$$(5) \quad \begin{array}{c|cccc} c_1 & b_1/\alpha_1 & 0 & \dots & 0 \\ c_2 & b_1 & b_2/\alpha_2 & & \vdots \\ \vdots & \vdots & \vdots & \ddots & \\ c_s & b_1 & b_2 & \dots & b_s/\alpha_s \\ \hline & b_1 & b_2 & \dots & b_s \end{array}$$

where $\alpha_i \neq 0$ for all $i = 1, \dots, s$. As usual, $c = (c_1, \dots, c_s)^t$ is given by $c = \mathcal{A}e$, with $e = (1, \dots, 1)^t \in \mathbb{R}^s$. Besides, we assume $b_i \neq 0$, $i = 1, \dots, s$, otherwise, if $b_i = 0$ for some i , then the s -stage DIRK method is equivalent to a reduced scheme with less than s stages.

Second order low-storage ASIRK- s A methods, satisfying different stability properties, were constructed in [8]. In particular, positivity of the coefficients was imposed to obtain Strong-Stability Preserving (SSP) methods (see, e.g., [14, 3], see too [4] and the references therein). Time-stepping processes with SSP methods ensure numerical preservation of qualitative properties like monotonicity and contractivity for convex functionals (norms, semi norms, entropy functions, etc.), positivity, etc. under nontrivial step size restrictions.

Observe that the explicit method $\tilde{\mathcal{A}}$ and the DIRK scheme \mathcal{A} in an ASIRK- s A method share the same weight vector b . Consequently, the requirement of positive weights for the SSP low-storage explicit method $\tilde{\mathcal{A}}$ implies that the low-storage DIRK scheme \mathcal{A} (5) should also have positive weights, that is, $b_i > 0$, $i = 1, \dots, s$.

It is well known that the order p of DIRK and Singly Diagonally Implicit RK (SDIRK) methods with positive weights must be $p \leq 6$ and $p \leq 4$, respectively (see[2, Th. 3.6.16 and Th. 3.6.17]). For SDIRK schemes, this barrier is obtained from [2, Lemma 3.6.12] or [5, IV.13, Lemma 13.12] and the fact that condition $\mathcal{A}c = c^2/2$, known as $C(2)$, does not hold. It is straightforward to prove that condition $C(2)$ does not hold either for methods (5), and thus the order p of low-storage DIRK schemes (5) with positive coefficients is always $p \leq 4$.

Several second order low-storage DIRK methods with positive weights were constructed in [8], but the unsuccessful attempts to construct third order low-storage DIRK schemes with positive weights made us think about the existence of these schemes.

The main result of the paper, that will be proven in the section 3, shows that these methods indeed do not exist.

Theorem 1. *There does not exist third order low-storage DIRK methods of the form (5) with $b_i > 0$ for all i .*

Thus, third order low-storage DIRK methods must contain negative weights. Example 1 below shows that, if we allow positive and negative weights, third order low-storage DIRK methods can be constructed (15).

On the other hand, there are second order low-storage DIRK methods with positive weights (see, e.g., the ones constructed in [8] and Proposition 4 below). Consequently, we have the following order barrier.

Corollary 1. *The class of low-storage DIRK methods of the form (5), with $b_i > 0$ for all $i = 1, \dots, s$, has the order barrier $p \leq 2$.*

As a result, the same order barrier holds for low-storage ASIRK- s A methods.

Corollary 2. *The class of low-storage ASIRK methods of the form (3)-(4) with $b_i > 0$ for all $i = 1, \dots, s$, has the order barrier $p \leq 2$.*

Theorem 1 and Corollary 1 extend a well known result for DIRK symplectic methods [15, pp. 100-102]. These schemes are of the form (5) with $\alpha_i = 2$ for all i ,

$$(6) \quad \begin{array}{c|cccc} c_1 & b_1/2 & 0 & \dots & 0 \\ c_2 & b_1 & b_2/2 & & \vdots \\ \vdots & \vdots & \vdots & \ddots & \\ c_s & b_1 & b_2 & \dots & b_s/2 \\ \hline & b_1 & b_2 & \dots & b_s \end{array}$$

and consequently they can be implemented in two memory registers. It can be proven that if the first order condition $b^t e = 1$ holds, then scheme (6) has also order two; furthermore, the two standard third order conditions are reduced to the following one

$$(7) \quad b_1^3 + b_2^3 + \dots + b_s^3 = 0.$$

Consequently, there are no low-storage DIRK schemes with $\alpha_i = 2$ and weights $b_i > 0$ for all $i = 1, \dots, s$, and the order barrier for these methods is $p \leq 2$.

Remark 1. In the context of SSP methods, the optimal second order s -stage SDIRK schemes are [3, 13]

$$\begin{array}{c|cccc} 1/(2s) & 1/(2s) & 0 & \dots & 0 \\ 3/(2s) & 1/s & 1/(2s) & & \vdots \\ \vdots & \vdots & \vdots & \ddots & \\ (2s-1)/(2s) & 1/s & 1/s & \dots & 1/(2s) \\ \hline & 1/s & 1/s & \dots & 1/s \end{array}$$

Observe that these optimal methods are symplectic SDIRK methods of the form (6) with positive weights. \square

The rest of the paper is organized as follows. In section 2, we study order conditions and some properties of low-storage DIRK schemes. Section 3 is devoted to low-storage DIRK schemes with positive weights; the proof of Theorem 1 is given in this section.

2 Low-Storage DIRK methods

In this section we study the order of low-storage DIRK methods of the form (5). Instead of dealing with the standard set of order conditions up to order three for the s -stage DIRK scheme (\mathcal{A}, b^t) , namely

$$(8) \quad \text{Order 1: } b^t e = 1, \quad \text{Order 2: } b^t c = 1/2, \quad \text{Order 3: } b^t c^2 = 1/3, \quad b^t \mathcal{A} c = 1/6,$$

we will make use of the fact that scheme (5) is a composition of the following 1-stage Runge-Kutta methods

$$(9) \quad \begin{array}{c|c} 1/\alpha_1 & \\ \hline 1 & \end{array} \quad \begin{array}{c|c} 1/\alpha_2 & \\ \hline 1 & \end{array} \quad \dots \quad \begin{array}{c|c} 1/\alpha_s & \\ \hline 1 & \end{array}$$

with step sizes b_1h, b_2h, \dots, b_sh , where

$$(10) \quad b_1 + \dots + b_s = 1.$$

Observe that (10) is the first order condition in (8). Then, we will use some results on order conditions for composite methods.

To see that (5) is a composite method, we simply have to observe that, from y_n , the numerical approximation at time t_n , the successive numerical approximations with the 1-stage methods in (9) with step sizes $hb_i, i = 1, \dots, s$, are given by

$$\begin{aligned} Y_1 &= y_n + h b_1 \frac{1}{\alpha_1} g(Y_1), \\ y^{(1)} &= y_n + h b_1 g(Y_1), \\ Y_2 &= y^{(1)} + h b_2 \frac{1}{\alpha_2} g(Y_2) = y_n + h b_1 g(Y_1) + h b_2 \frac{1}{\alpha_2} g(Y_2), \\ y^{(2)} &= y^{(1)} + h b_2 g(Y_2) = y_n + h b_1 g(Y_1) + h b_2 g(Y_2), \\ &\vdots \\ Y_s &= y^{(s-1)} + h b_s \frac{1}{\alpha_s} g(Y_s) = y_n + h \sum_{i=1}^{s-1} b_i g(Y_i) + h b_s \frac{1}{\alpha_s} g(Y_s), \\ y^{(s)} &= y^{(s-1)} + h b_s g(Y_s) = y_n + h \sum_{i=1}^s b_i g(Y_i). \end{aligned}$$

A simple comparison shows that the values Y_i in the above expression agree with the internal stages Y_i of the scheme (5), and the value $y^{(s)}$ agrees with y_{n+1} , the numerical approximation at time t_{n+1} of the scheme (5). Using the composition methods notation (see, e.g., [6]), one h -step with scheme (5) is equivalent to

$$\Psi_h = \Phi_{b_s h}^{1/\alpha_s} \circ \Phi_{b_{s-1} h}^{1/\alpha_{s-1}} \circ \dots \circ \Phi_{b_2 h}^{1/\alpha_2} \circ \Phi_{b_1 h}^{1/\alpha_1},$$

where Φ_h^θ denotes one h -step with the θ -method

$$\begin{array}{c|c} & \theta \\ \hline & 1 \end{array}$$

Remark 2. If some weight b_i is negative, a backward integration step is given in the composite method. For ODEs that arise from a spatial discretization of some PDEs that are ill-posed for negative times, e.g., dissipative systems like those involving the Laplacian operator, backward integration steps are undesirable [6, p. 82]. \square

Remark 3. Recall that, in the geometric numerical integration context, the composition of a given basic one-step method (and eventually, its adjoint scheme) with different step sizes is considered [6, II.4]. However, in this paper we deal with the composition of different 1-stage RK methods with different step sizes. \square

The main result of the paper can be obtained from the order barrier of scheme (5) for the linear scalar test equation $y'(t) = \lambda y(t)$. For an s -stage RK method (\mathcal{A}, b^t) , it is well known that y_n and y_{n+1} , the numerical approximations at times t_n and $t_n + h$, respectively, satisfy

$$y_{n+1} = R(h\lambda)y_n,$$

where $R(z)$ is the stability function of the method, defined by

$$R(z) = 1 + zb^t(I_s - z\mathcal{A})^{-1}e.$$

The RK scheme has linear order p if and only if

$$R(z) - \exp(z) = \vartheta(h^{p+1}).$$

The set of order conditions to achieve linear order p can be obtained from the Taylor expansion of the stability function given by

$$R(z) = 1 + (b^t e) z + (b^t c) z^2 + (b^t \mathcal{A}c) z^3 + (b^t \mathcal{A}^2 c) z^4 + \dots.$$

Thus, a RK method has linear order 3 if and only if the following conditions are satisfied

$$b^t e = 1, \quad b^t c = 1/2, \quad b^t \mathcal{A}c = 1/6.$$

A comparison with (8) shows that the third order condition $b^t c^2 = 1/3$ is not required in the linear case. See, e.g., [5] for details.

As we have pointed out above, an s -stage low storage DIRK method (5) can be considered as a composition of s 1-stage RK methods. Then, it is straightforward to prove that the stability function of the composite method is given by

$$R(z) = R_1(b_1 z) R_2(b_2 z) \cdots R_s(b_s z),$$

where $R_i(z)$ denotes the stability function of each 1-stage RK scheme in (9).

The order conditions for the composition of rational approximations to $\exp(z)$ have been studied in [9]. For each 1-stage RK method in (9) the stability function $R_i(z)$ is a rational approximation to $\exp(z)$ and, consequently, we can use the results in [9]. In particular, Theorem 2 in [9] gives the following result for the case $p = 1$.

Theorem 2. Assume that $R_i(z)$, the stability function of each 1-stage scheme in (9), satisfies

$$(11) \quad R_i(z) - \exp(z) = \beta_{i,2} z^2 + \beta_{i,3} z^3 + \vartheta(z^4).$$

Then,

1. The composite method (5) has order 2 if and only if

$$(12) \quad b_1^2 \beta_{1,2} + \cdots + b_s^2 \beta_{s,2} = 0.$$

2. The composite method (5) has linear order 3 if and only if conditions (12) and

$$b_1^3(\beta_{1,2} - \beta_{1,3}) + \cdots + b_s^3(\beta_{s,2} - \beta_{s,3}) = 0$$

hold.

Using this theorem, we can prove the following result.

Proposition 1. Consider a low-storage DIRK method (5) such that the first order condition, $b^t e = 1$, holds. Then,

1. The low-storage DIRK method (5) achieves order 2 if and only if

$$(13) \quad b_1^2 \left(\frac{1}{\alpha_1} - \frac{1}{2} \right) + \cdots + b_s^2 \left(\frac{1}{\alpha_s} - \frac{1}{2} \right) = 0.$$

2. The low-storage DIRK method (5) achieves linear order 3 if and only if conditions (13) and

$$(14) \quad b_1^3 \left(\frac{1}{12} + \left(\frac{1}{\alpha_s} - \frac{1}{2} \right)^2 \right) + \cdots + b_s^3 \left(\frac{1}{12} + \left(\frac{1}{\alpha_s} - \frac{1}{2} \right)^2 \right) = 0$$

hold.

Proof. For each 1-stage method in (9), the coefficients $\beta_{i,2}$ and $\beta_{i,3}$ in (11) are given by

$$\beta_{i,2} = \left(\frac{1}{\alpha_i} - \frac{1}{2} \right), \quad \beta_{i,3} = \left(\frac{1}{\alpha_i^2} - \frac{1}{6} \right).$$

Using part 1 in Theorem 2, we obtain part 1. For part 2, we simply have to use part 2 in Theorem 2 and write $\beta_{i,2} - \beta_{i,3}$ as

$$\beta_{i,2} - \beta_{i,3} = \frac{1}{\alpha_i^2} - \frac{1}{6} - \left(\frac{1}{\alpha_i} - \frac{1}{2} \right) = \frac{1}{12} + \left(\frac{1}{\alpha_i} - \frac{1}{2} \right)^2.$$

□

□

For DIRK symplectic methods (6), we have that $\alpha_i = 2$ for all i . Thus, from Proposition 1, we recover the following well known results.

Corollary 3. Consider a first order DIRK symplectic method (6). Then

1. It also achieves order 2.
2. The method has linear order 3 if and only if condition (7) holds.

From Proposition 1 we can also obtain some other interesting properties for second order low-storage DIRK methods. The following result, whose proof is straightforward, considers second order low-storage DIRK methods with $\alpha_i = \alpha$ for all i .

Proposition 2. Consider a first order low-storage DIRK method (5) with $\alpha_i = \alpha$ for all i . Then the method has order 2 if and only if $\alpha = 2$.

Consequently, second order low-storage DIRK methods with $\alpha_i = \alpha$ for all i are symplectic DIRK schemes (6).

From equation (13) we can also get conditions on coefficients α_i for second order low-storage DIRK methods.

Proposition 3. Consider a second order low-storage DIRK method (5). Then just one of the following statements holds:

1. The coefficients $\alpha_i = 2$, $i = 1, \dots, s$.
2. There exist indexes i and j such that $\alpha_i > 2$ and $\alpha_j < 2$.

Proof. If there exists an index j_0 such that $\alpha_{j_0} \neq 2$, then equation (13) and the fact that $b_i \neq 0$ for all i , imply that there must be indexes i and j such that $\alpha_i > 2$ and $\alpha_j < 2$. \square \square

For example, for the DIRK part of the second order low-storage ASIRK methods considered in [8], named as ASIRK-LSe(3,2) ASIRK-LSs(3,2), ASIRK-LS(3,2), the coefficients α_i are given by

Method	α_1	α_2	α_3
ASIRK-LSe(3,2)	1	3.54762	1
ASIRK-LSs(3,2)	1	3.76587	1
ASIRK-LS(3,2)	4.29529	2.41085	1

We finish the section pointing out that positivity of the weights b_i has not been assumed in this section. This special case is considered in the next one.

3 Low-storage DIRK methods with positive weights

By using Proposition 1 we can prove Theorem 1 stated in Section 1.

Proof. of Theorem 1 In (14), the coefficients b_i^3 are multiplied by positive terms. Thus, if $b_i > 0$, for $i = 1, \dots, s$, equation (14) does not have any solution. Consequently, there are no low-storage DIRK methods with linear order 3. \square \square

If we allow positive and negative weights, third order DIRK low-storage methods can be constructed.

Example 1. The following coefficients correspond to a third order 4-stage low-storage DIRK method.

$$\begin{aligned}
 (15) \quad & b_1 = 0.432015223352125, & b_2 = 0.962439982373961 \\
 & b_3 = -1.033807280340946, & b_4 = 0.639352074614861 \\
 & \alpha_1 = 2.946745495781493, & \alpha_2 = 1.341225644531434 \\
 & \alpha_3 = 4.283870450039035, & \alpha_4 = 1.400968981668695.
 \end{aligned}$$

\square

With Propositions 1 and 3 it is straightforward to construct 2-stage second order low-storage DIRK schemes with positive weights (see too [9, Theorem 3]).

Proposition 4. *The family of 2-stage second order low-storage DIRK methods with positive weights is given by*

$$(16) \quad \begin{array}{c|cc} b_1/\alpha_1 & b_2/\alpha_1 & \\ \hline b_1 + b_2/\alpha_2 & b_1 & b_2/\alpha_2 \\ \hline & b_1 & b_2 \end{array}$$

with either $\alpha_1 = \alpha_2 = 2$, $b_1 = \theta$ and $b_2 = 1 - \theta$, with $\theta \in (0, 1)$, or

$$(17) \quad b_1 = \frac{\delta}{1 + \delta}, \quad b_2 = \frac{1}{1 + \delta},$$

where

$$\delta = \sqrt{\frac{\alpha_1(2 - \alpha_2)}{\alpha_2(\alpha_1 - 2)}}.$$

Proof. By Proposition 1, to achieve second order, conditions

$$b_1 + b_2 = 1, \quad b_1^2 \left(\frac{1}{\alpha_1} - \frac{1}{2} \right) + b_2^2 \left(\frac{1}{\alpha_2} - \frac{1}{2} \right) = 0,$$

must be imposed. By Proposition 3, either the coefficients are $\alpha_1 = \alpha_2 = 2$ or they must satisfy $(\alpha_1 - 2)(\alpha_2 - 2) < 0$. For the first case, as the second order condition is fulfilled, it is enough to impose $b_1 + b_2 = 1$. For the second case, from the second order condition and the positivity of the weights, we get $b_1 = b_2\delta$, and the first order condition gives (17). \square

Consequently, there exist second order low-storage DIRK schemes; some 3-stage second order low-storage DIRK methods can be seen in [8].

Remark 4. For $\delta = 1$, we have that $b_1 = b_2 = 1/2$ and

$$\frac{1}{\alpha_1} = 1 - \frac{1}{\alpha_2}.$$

In this case, method (16) is the composition of a 1-step method and its adjoint scheme [6, p. 136]. \square

4 Conclusions

In this paper we have obtained an order barrier for low-storage DIRK methods with positive coefficients. These schemes are the implicit part of the low-storage implicit-explicit methods studied in [8]. We have considered the van der Houwen's low-storage format but there are other types of low-storage explicit methods (see, e.g., [17, 11, 12]), that might be extended to implicit-explicit pairs; a different class of low-storage implementations might break the order barrier $p \leq 2$. Consequently, in order to obtain high order low-storage implicit-explicit methods, either more memory registers or other low-storage implementations should be considered. This will be the object of a forthcoming work.

Acknowledgments

We thank the reviewers for their helpful comments and remarks on the paper.

Supported by Ministerio de Economía y Competitividad, project MTM-2014-53178-P.

References

- [1] M. Calvo, J.M. Franco, J.I. Montijano, and L. Rández. On some new low storage implementations of time advancing Runge-Kutta methods. *J. Comput. Appl. Math.*, 236(15):3665–3675, 2012.
- [2] K. Dekker and J. G. Verwer. *Stability of Runge-Kutta methods for stiff nonlinear differential equations*. CWI Monographs, North-Holland, 1984.
- [3] L. Ferracina and M.N. Spijker. Strong stability of singly-diagonally-implicit Runge-Kutta methods. *Appl. Numer. Math.*, 58(11):1675–1686, 2008.
- [4] S. Gottlieb, D. I. Ketcheson, and C. W. Shu. *Strong Stability Preserving Runge-Kutta and Multistep Time Discretizations*. World Scientific Publishing Company, 2011.

- [5] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II*. Springer Series in Computational Mathematics 14. Springer, Berlin, 1991.
- [6] E. Hairer, C. Lubich, and G. Wanner. *Geometric numerical integration: structure-preserving algorithms for ordinary differential equations*, volume 31. Springer Series in Computational Mathematics, 2006.
- [7] N. Happenhofer, O. Koch, and F. Kupka. IMEX Methods for the ANTARES Code. *ASC Report*, 27, 2011.
- [8] I. Higuera and T. Roldán. Construction of additive semi-implicit Runge-Kutta methods with low-storage requirements. *J. Sci. Comput.*, 67(3):1019–1042, 2016.
- [9] A. Iserles. Composite exponential approximations. *Math. Comput.*, 38(157):99–112, 1982.
- [10] C.A. Kennedy, M.H. Carpenter, and R.M. Lewis. Low-storage, explicit Runge-Kutta schemes for the compressible Navier-Stokes equations. *Appl. Numer. Math.*, 35(3):177–219, 2000.
- [11] D.I. Ketcheson. Highly efficient strong stability-preserving Runge-Kutta methods with low-storage implementations. *SIAM J. Sci. Comput.*, 30(4):2113–2136, 2008.
- [12] D.I. Ketcheson. Runge-Kutta methods with minimum storage implementations. *J. Comput. Phys.*, 229(5):1763–1773, 2010.
- [13] T. A. Kocsis and A. Németh. Optimal second order diagonally implicit SSP Runge–Kutta methods. *arXiv:1409.8583v1*, 2014.
- [14] J. F. B. M. Kraaijevanger. Contractivity of Runge-Kutta Methods. *BIT*, 31:482–528, 1991.
- [15] J. M. Sanz-Serna and M. P. Calvo. *Numerical Hamiltonian Problems*. Chapman & Hall, London, 1994.
- [16] P. J. Van Der Houwen. *Construction of integration formulas for initial value problems*. North Holland, 1977.
- [17] J. H. Williamson. Low-storage Runge-Kutta schemes. *J. Comput. Phys.*, 35(1):48–56, 1980.
- [18] X. Zhong. Additive Semi-Implicit Runge–Kutta Methods for Computing High–Speed Nonequilibrium Reactive Flows. *J. Comput. Phys.*, 128(1):19–31, 1996.